Central Fault Tolerance For Dual Database Server Real Time System

Rana Dhia'a Abdu-Aljabar*& Asia Ali Salman**

Received on: 11/2/2009
Accepted on: 11/3/2010

Abstract

The aim of this article is to find an efficient method to detect fault in dual database server which is working on critical environment real time system (such as power and water distributed environment).

In traditional dual database server the fault tolerance is embedded in each server. So when there is any defectiveness, each server try to uncover the error in separate way. This led to increase the load on each server and job lateness.

This paper proposes a central fault tolerant method for dual database server through a centralized control, so that the fault will be more controlled and manipulated and the load will be less in each server since problems detection and correction will not depend on dual server but it will be centralized.It showed practically how the dual server worked under fault conditions and critical environment such as distributed real time systems.

Keywords: Fault tolerance, Dual Server

مصحح اخطاء لخادم قاعدة بيانات ثنائي في انظمة الوقت الحقيقي

الخلاصة

الذي من هذا البحث هو لإيجاد طريقة كفوءة لتحسس الخطأ في قواعد البيانات ثنائية ألمُخدَم يعمل في بيئة حرجة مثل انظمة الوقت الحقيقي مثل مشاريع توزيع الطاقة الكهربائيـــة والمـــاء. الهدف في ثنائي ألمُخدَم التقليدي يكون عادة متحمل العيب متضمناً في كل خادم ، فعند وجود أي خلل يقوم كل خادم بمحاولة كشف الخطأ بشكل منفصل، مما يؤدي إلى زيادة حمل كل خادم ومن ثم تأخير في عملهما.في هذه الدراسة نطرح منهج متحمل العيب لثنائي المخدم من خلال السيطرة المركزية وذلك لسيطرة ومعالجة أكثر للعيب ولتقليل الحمل علـــى الخـــادمين بمـــا ان مشـــاكل الاكتشاف والتصحيح سوف لن تكون معتمدة على ثنائي ألمُخدَم وإنما ســـوف تكـــون مركزيـــة. وسنعرض عمليا كيفية عمل هذا النظام تحت الظروف الخاطئة وفي بيئة حرجـــة مثـــل انظمـــة توزيعات الوقت الحقي

1- Introduction

The failure of a server of the network may cause a network service to fail to respond ongoing request and to no longer be available to new request [1].

A central fault tolerance method is used for Dual database server to increase the availability and reliability of network services. Reliability and availability are essential characteristics of computer systems operation. A runtime monitoring system contributes for improving reliability and availability, respectively, by continuous failure detection and by reducing time to diagnose failures [2].

* Information Engineering College, University of Al_Nahrain / Baghdad
** Computer Sciences Department, University of Technology/ Baghdad

https://doi.org/10.30684/etj.28.12.12
2412-0758/University of Technology-Iraq, Baghdad, Iraq

Dual database means that there are two servers both of them have the same Database files and each one is on a private computer. Dual server are connected through the network using a specific network communication technique, any client attend to deal with the database files it must notify the dual server that it's exist so its request go through the network and gets the information it needs [3].

The causes of service failures can be divided into three types: network faults, client faults and server faults [4]. Our work focuses on recovering from faults that occur within a ''server room''.

Fault tolerance is a crucial design consideration for mission critical distributed real-time systems (DRT), which combine the real-time characteristics with the dynamic characteristics of distributed platforms. DRT systems present unique challenges for traditional fault tolerance approaches because of their scale, heterogeneity, real-time requirements, and other characteristics.

This paper firstly mention the related work, those work that search in the same field but in different way, then describes system structure which represents the method used to implement this system for hardware and software. Then draw the schemes for dual server and Fault tolerance control which emerge how they distributed in the system platform and there connections. After that, demonstrated the system performance under the correct and fault conditions. Finally, it summarizes the conclusion of this work.

2- Related Work

Fault tolerance for distributed systems is an active area of research and many projects have made significant contributions. In [5] they use an algorithm that achieves N - 1 fault-tolerant resiliency for N-server video-on-demand systems. In [6], they design and implement a dual-module sun workstation redundant system which fault-tolerant computer system has the performance that if there are some troubles occurs the system itself has the capacity to find and correct or eliminate those troubles, and ensure the whole system running normally. This is followed by using Dual redundancy server (primary and secondary server), if the primary server fails the client's requests data from the second server [3, 7].

Another more recent solution is that the fault tolerance is made possible by the partitioned architecture of the system and data redundancy control actions include restoration of lost data sets in a single server using redundant data sets in the remaining servers [8], and in [9] it provides a reliability monitoring scheme for fault tolerance control systems.

3- System Structure

The system that this paper worked on is shown in figure (1) which is discussed in details in [3]. This system has been used in Iraqi national Control Center. This paper suggests figure (2) to improve the old system.

The proposed system is implemented in special environment that had chosen to implement and test this job which can be summarized as follow:

1- Number of personal computers is six.
2- The operating system is Windows XP.
3- Interconnected by an Ethernet Network with 100 Mbps speed and the topology is star connection.
4- The programming language is Visual C++ (MFC and Win32 application).
5- The database was build, at first, using visual FoxPro then converted to C++ files using a specialized program written for this purpose.

To discuss the proposed system it will be partitioned into two parts, Dual Server and Central Fault tolerance.

4- Dual Server Scheme

This paper design a backup dual database server as shown in (figure 3) to increase the reliability of the system in order to keep control the access to the database and to deal with fault tolerance of critical tasks.

The Dual server connection is done by sending their locations to each other (see figure 3). This connection is very important to make each server know the state of the other. It uses a handshaking message, which is continually sent between them, so it can handle the failure situation when occurs in any server.

When the clients run they start connect to the dual server in order to access database files which is controlled by dual server, each client will either read, write or update the information exist in dual database files [3].

When the Dual server performing their work in real time processing systems such as power or water distributed environment they make them overloaded. Therefore to lighten the load on dual server, this paper suggests a central fault tolerance that discuss below.

4- Center Fault Tolerance Scheme

To give the Fault Tolerance program (FTP) full capability to control the dual server computers, there must be a program work as a service to it on each Server's computer and its backup. That service is a special kind of win32 process that makes it a good choice for software that needs to start automatically and run constantly in the background, whenever the system is up [10]. This program or service is called Fault Tolerance Service (FTS) (see figure 4). It's always able to receive and do any orders come from Fault Tolerance program (FTP) and resend a message to inform it if the order is done successfully or not, and the state of the Server during its work.

5- Demonstration of System Performance

5.1-Demonstration of Central FTP

The FTP has two phases starting phase and running phase. And it has a very important file which is called Server_Information file (see figure 5). This file is responsible for specifying the dual server locations and their backups and FTP backup.

When the FTP files Backup button pushed in Server_Information file form, another form will represented (see figure 6) which asks for a computer name to store the necessary file of FTP that need them when it failed. When the FTP start running, it first read the Server_Information file and search for

FTP backup files location and servers information if it didn't find any of them, it will informed the operator by a dialog box to assigning them first.

The Server_Information file has only Change tools for modification because it is fixed file has only two servers as processes so it has no add or delete tools. In change tools form (see figure 7) the operator can write or change the default and backup locations and computers.

At the starting phase it has two steps the Connection step and Loading step (see algorithm 1). When the Run button pushed from the main inter face form (see figure 8) the FTP read the Server_Information file then begin to connect with each default and backup computers by two connections. One for send and one for receive (this technique make the connection faster) using named pipe protocol. Then it sends messages to FTS for loading the dual server in correct place. If the booting success, the Fault monitoring function will run (see figure 9).

The fault monitoring has several functions which monitored on the dual server and the FTP itself hardware and software.

In reality, however, it is not practical to monitor every state variable in a network. As a result, knowledge on a certain set of states is inferred based on the observables. On the other hand, a control action, in response to a state transition such as an occurrence of a server failure, must wait until a process of diagnosing the failure state is complete. The time required for diagnosis is assumed to be a random variable and the outcome of the diagnosis usually has some degree of uncertainty as well [8].

The fault monitoring on Dual Server will discussed here, It have two types of functions; first responsible for software failure and the second for hardware failure. The FTS is responsible for software failure monitoring if anything happened it tries to resolve it by re load the software again (see Algorithm 2). It tries to do that twice time if not success it tries to reload it from backup location. Because the software may have damaged for any reason if it also fail it sends a message to FTP which it declares this Server as a damaged server, so that the other server will take a handle till the FTP load this server on the backup computer and changes the necessary software connections to that computer.

The FTP is responsible on hardware fault monitoring if any error happened like computer failure or wire crosscutting and so on, it takes the same operation that mentioned earlier (see algorithm 3).

Each fault happen, the operator informed about it immediately ,(see figure 10) where the time of detecting network error is less than the time taken by the dual server which was (18766) millisecond (see figure 18), while in the proposed system is (14577) millisecond as shown in (figure 10).

In State Monitoring (see either figure 8, 9 or 10) there are three buttons. First one is the *Current State button* which is responsible to show the current configuration of the servers (see figure 11). The second one is the *Computer State button*, informed the operator if there is any computer out

of the system (see figure 12). The last one which is the *Fault Archiving button* it stored all fault happened in the system, its time and what the FTP done about it (see figure 13).

The fault monitoring function on FTP itself is done by two special FTS. After booting are success the FTP chooses two FTS depending on Server_Information file and declares one of them which it is lies on the same computer with FTP as FTS Recovery (FTSR) and the other which is lies on the FTP backup computer as Backup FTS Recovery (BFTSR). Each of them takes different action when FTP is damaged.

When FTP damaged, all the FTS detect that because the pipes are broken. Two of them which are chosen by FTP as a recover services will take a handle of that situation (see algorithm 4, 5). Different kind of messages will be translated between them to know who will take an action at that moment. The priority usually is to FTSR.

When the FTSR detects the FTP damaged, it tries to reload it from the same location. If it fails, it retries to load it from backup location. Taking in consideration the last state of the FTP i.e., before it has been damaged.

At that time the BFTSR sends a message to FTSR, and asks it if it is active. If the answer is yes, BFTSR takes no farther action. If FTSR does not respond for any reason, for example, its computer is damaged; the BFTSR will take an action. It tries to reload the FTP from default or backup location, on backup computer and on the last state that it was before the damage.

5.2- Demonstration of Dual Server under Central FTP Methodizing

In dual server each server has almost three tasks, algorithm (6) shows the dual server functions under central FTP:

1-When dual server start, they establish communication path between them by sending a handshaking message to each other, using Mailslot connection technique, that it exists and starts successfully if it is established then statues information sends from one server to another server. A living message is continuously send between the dual server using named pipes connection technique to make each server notes quickly if one of them is disconnected from the other one) suppose that their names are (dual_server1) and (dual_server2).

If at starting, both of them start successfully then they will stay waiting for any incoming client that want to connect to them (see figures 14, 15). If any one came then communication path is established between the dual server and this new client(s) by named pipe technique. If this client wants only to perform a read operation from the database files it will logically deals with the first server that it is connect to it. But if the operation is to update (write) database it will deal with the dual server (see figure16, 17).

2- Suppose that if dual_server2 is fail to start, the dual_server1 could deals with this event since there is a communication path between them till a central FTP take a suitable action. In this situation the dual_server1 will blocked all connected clients for a period then it returned them to the

work again, to make them notice that it is the only existing server (the master server) and they must access only to it in reading, writing or updating database files (see figure18).

Note that the same operation is done if the dual_server1 is stopped and the dual_server2 is still in service.

3- If the second server (dual_server2) is returned to work by FTP (see figure 18), the master server (dual_server1) would notice this event and must block (notify) the clients and synchronize its database files with the dual_server2 database (make a fresh copy of the database files for the (dual_server2)). Then make the clients return to work again (see figure 20).

While From the client side, each client has three tasks (Note that the following description is about only one client and it is the same for all other clients):

1. When client start and find the dual server is working the client(s) will connect to them and perform its tasks on the dual server database files (see figure16, 17).
2. When client(s) is working with dual server normally, then one of the dual servers (suppose the (dual_server2)) fail, the client(s) will disconnected from the (dual_server2) depends on the message came from (dual_server1) (see figure18). It will perform all its work (dealing with database files) by the (dual_server1) (which is the master server now).
3. Suppose the (dual_server2) returned to work by FTP, the master server (dual_Server1) would notice that event and must block (notify) the client(s). The client(s) stay waiting until master server synchronize database of the (dual_server2) and send message to the client(s) telling them that the (dual_server2) is working (see figure 20). The Client(s) will work with dual server; means the update database files performed in the (dual_server1) as well as in the database files in the (dual_server2), while the read operation will stay with the (dual_server1).

6- Conclusions

Providing central fault tolerance in distributed real-time systems involves carefully crafting and integrating techniques to not only meet reliability requirements, but also to match the characteristics of the systems. The work that reported in this paper not only provides a practical case study of inserting a central fault tolerance into a complex system of interacting components (dual-server) , but also provides several significant results in fault tolerance for distributed real-time systems.

First, it improves the performance of dual database server by making the load less in each server and the fault become more controlled and manipulated. Since the results show that the central system detects for example the network error faster than the time taken by the old dual server.

Secondly, the use of active replication to mask faults made the system very rapid recovery, so it becomes nearly continuous availability which it is mach the characteristics of specific systems.

Thirdly, it improves the design and implementation of fault tolerance by making it applicable not for dual server only but also to larger classes of systems and problems.

7- References

[1] Navid Aghdaie and Yuval Tamir, "Fast Transparent Failure for Reliable Web Service", Proceedings of the international conference on Parallel and Distributed Computing and Systems, Marina del Rey, California, P.P. 757-762, November 2003.

[2] Sérgio Ricardo Rota; Jorge Rady de Almeida Jr., "Run-Time Monitoring for Dependable Systems: an Approach and a Case Study", Proceedings of the 23rd IEEE International Symposium on Reliable Distributed Systems, P.P. 41-49, Oct. 2004.

[3] Asia Ali, Intisar Shaded, Khalid Tourky Atah, "Design and Implementation of Dual Database Server and Clients", Scientific Journal, Speciliced and well-Knit Published by University of Technology, Baghdad, Iraq, Vol. 8, No. 1, 2008.

[4] Navid Aghdaie, "*Transparent Fault-Tolerant Network Service Using Off the Shelf Component",* PHD thesis, University of the California, Los Angeles, 2005.

[5] Ing-Jye Shyu; Shiuh-Pyng Shieh, "*A Distributed Fault-Tolerant Design for Multiple-Server VOD Systems*", Journal of Multimedia Tools and Applications 8, P.P. 219–247, 1999.

[6] Yi Zhuang; Yang Lu; Bo Zhu; Min Zhu, "*Design and implementation of a dual-server fault-tolerant real-time processing system*", Proceedings of the 3rd World Congress on Intelligent Control and Automation, volume 5, P.P.3618 – 3622, 2000.

[7] CitectSCADA version 6.0, "CitectSCADA_Tech_Overvie.pdf" ©copyright 2004, Citect Corporation.

[8]- N. EvaWu, Matthew C. Ruschmann, and Mark H. Linderman," "*Fault-Tolerant Control of a Distributed Database System*",Journal of Control Science and Engineering, Volume 2008, Article ID 310652, P.P. 145-157, 2008.

[9] Hongbin Li; Qing Zhao; Zhenyu Yang, "*Reliability Monitoring of Fault Tolerant Control Systems with Demonstration on an Aircraft Model*" Journal of Control Science and Engineering, Volume 2008, Article ID 310652, P.P. 135-144, 2008.

[10] Microsoft Developer Network (MSDN) Library, October 2001.

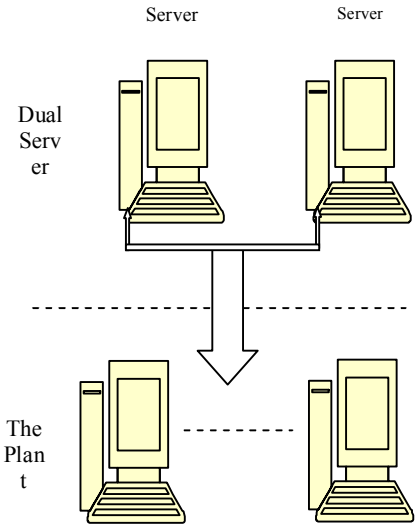

Figure (1) The old System Structure

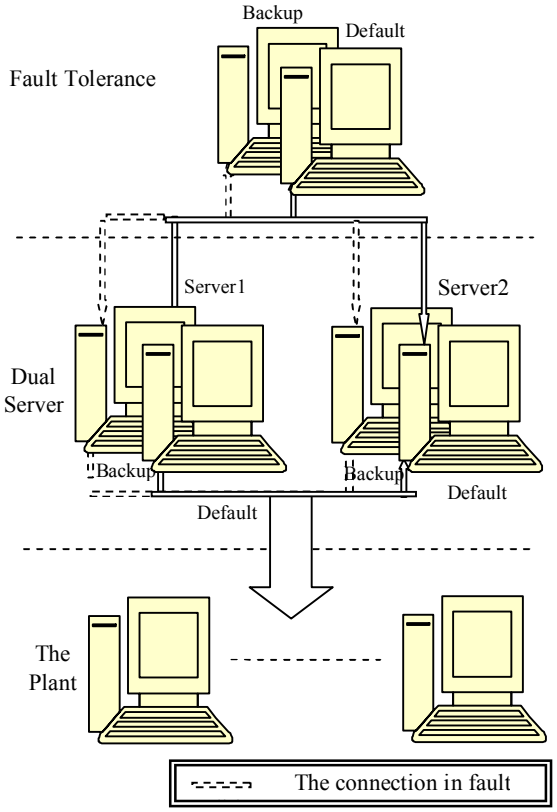

Figure (2) proposed System structure

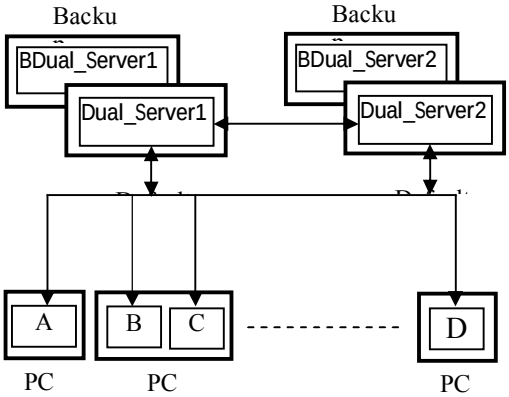

Figure (3) Dual_Server Scheme where A,B,C,and D are Clients

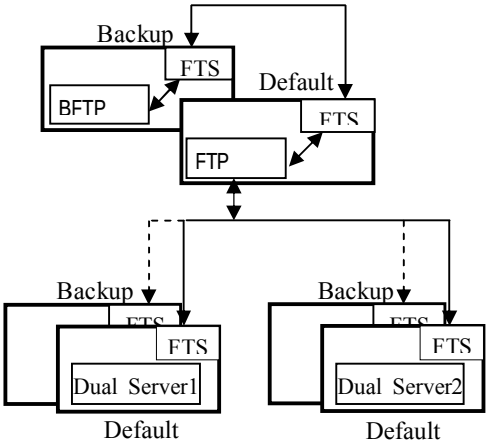

Figure (4) Fault Tolerance Scheme

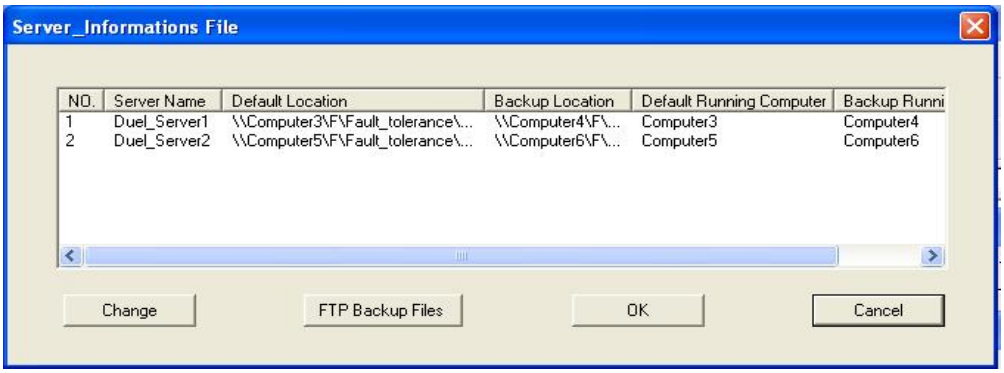

Figure (5) Server_Information File of FTP

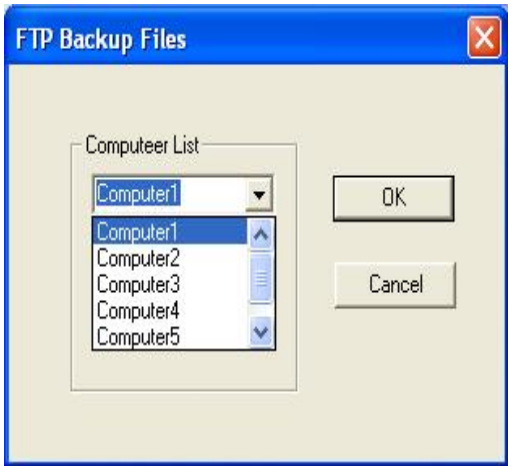

Figure (6) FTP Backup File Form

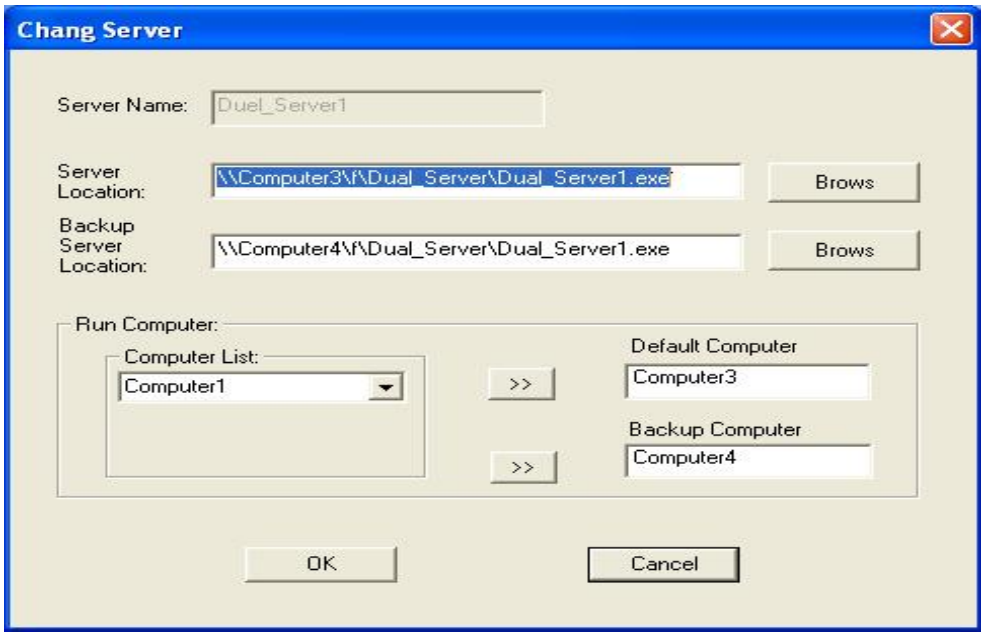

Figure (7) Change Form of Server_Information File

1-Input :Server_Information File
2-Output: Run Monitoring function
3-Read Server_Information File
4-Connect Read and write Pipe connection with all FTS on all computers
5-Loop for twice [one for Dual_Server1 and one for Dual_Server2]
6- Run (ServerName) on its default computer
7- If (Not Success) then
Run (ServerName) on its backup computer
8- End Loop
9-If (Booting Success) then
Run Monitoring function
10- End //end of algorithem

Algorithem 1 FTP Funcion

```
1-Input: ServerName
2-Output: re-run ServerName on FTS computer
3-re-run (ServerName) on FTS computer
4- If ((ServerName) fail) then
5-      Loop for twice
6-              Re-run the (ServerName)
7-              If (Success) then
                        Goto step 11
8-      End Loop
9-      If (Not Success) then
                Send a message to FTP
10- End If      // end of If in step 4
11-End          //end of algorithem
```

Algorithem 2 FTS Function

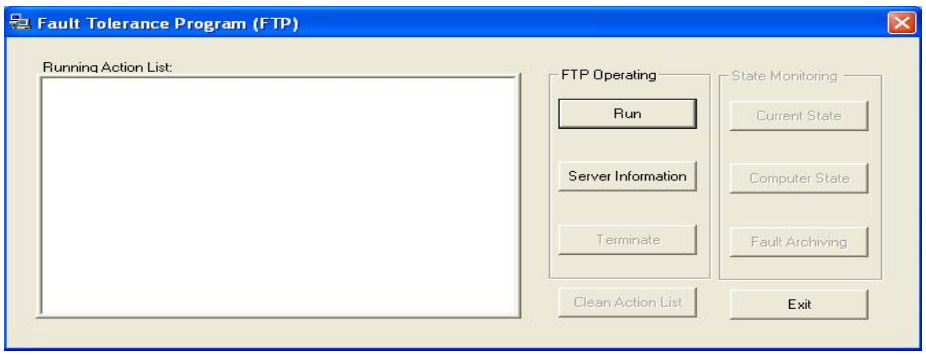

Figure (8) Main Interface of FTP

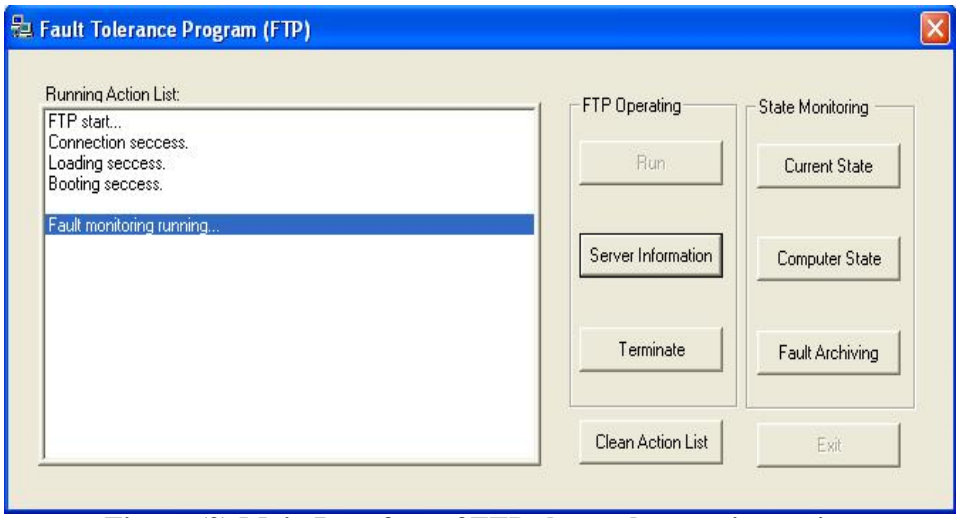

Figure (9) Main Interface of FTP shows the running action

```
1-input: ServerName
2-output:Re-Connect the Pipe
3-If (Pipe Disconnected) then
        Run (ServerName) on its backup computer
        Write the fault on Fault_Archiving File
4-      Loop
5-              Re_Connect the Pipe
6-              If (Pipe Connection Success) then
                        Goto step 9
```

Algorithem 3 FTP Function

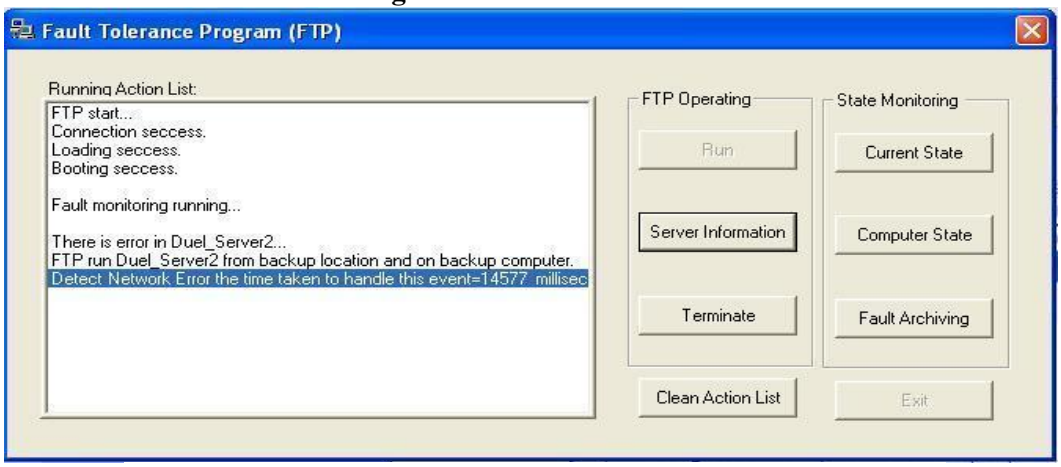

Figure (10) Instant fault represented immediately in Action

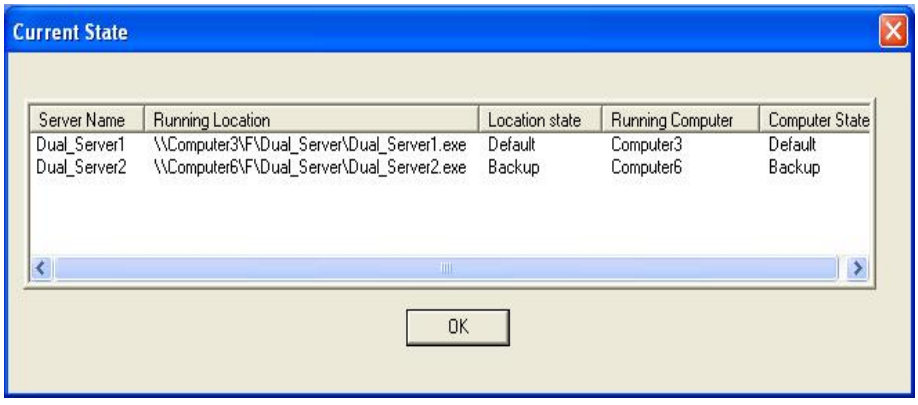

Figure (11) Current State Form

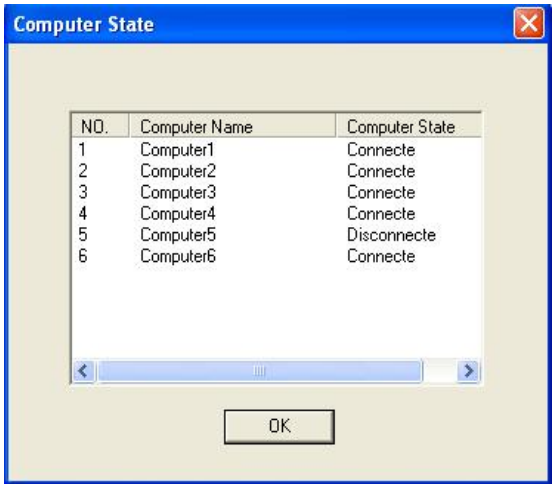

Figure (12) Computer State Form

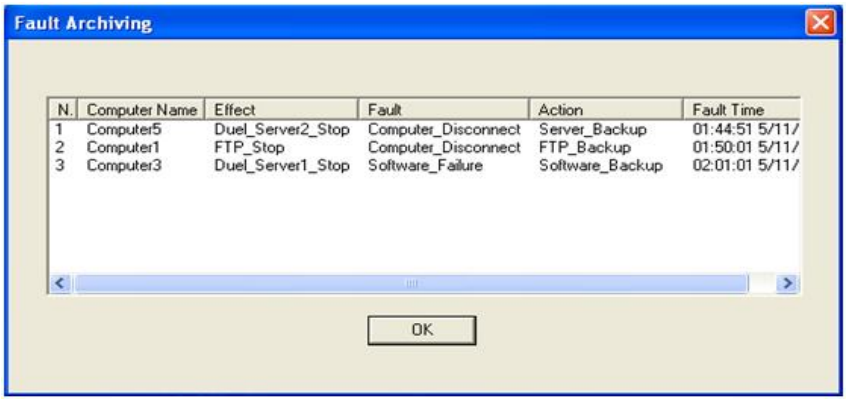

Figure (13) Fault Archiving Form

Figure (14) Dual Server1

```
1-input: none
2-output: re-run FTP
3-If (Pipe Disconnected) then
        Re-Run FTP on its Default computer from default location
4-      If (Not Success) then
                Re-Run FTP on its Default computer from backup
        location
5-End If        //end of If in step 3
6-End  //end of algorithem
```

Algorithem 4 FTSR Function

```
1-input: none
2-output: re-run FTP
3-If (Pipe Disconnected) then
        Send a message to FTPR (ARE YOU A LIVE)
4-      Read a message (wait for a period of time)
5-        If (Read Nothing) then        /*it means FTPR has a problem
                                        and BFTSR must take an action*/
                Re-Run FTP on its Backup computer from default
        location
6-              If (Not Success) then
                        Re-Run FTP on its Backup computer from
                backup location
7-      End If  //end of If in step 5
```

Algorithem 5 BFTSR Function

1-Input: none
2-Output: run dual server
3- Initialize the variables and create connection method used to send information to other server
4-Create the following operations:

A. Watch_DualDB (watch the other dual server)
B. DualDB_Moniter_client(Wait for new clients)
C. DualDB_Client(keep connection to the other server)
D. If(Not(Shutdown or stop or restart server)) then go to 4
E. Perform all un Initialize the variables and terminate dualDB_Client terminate Watch_DualDB

Algorithm (6) Dual Server Functions

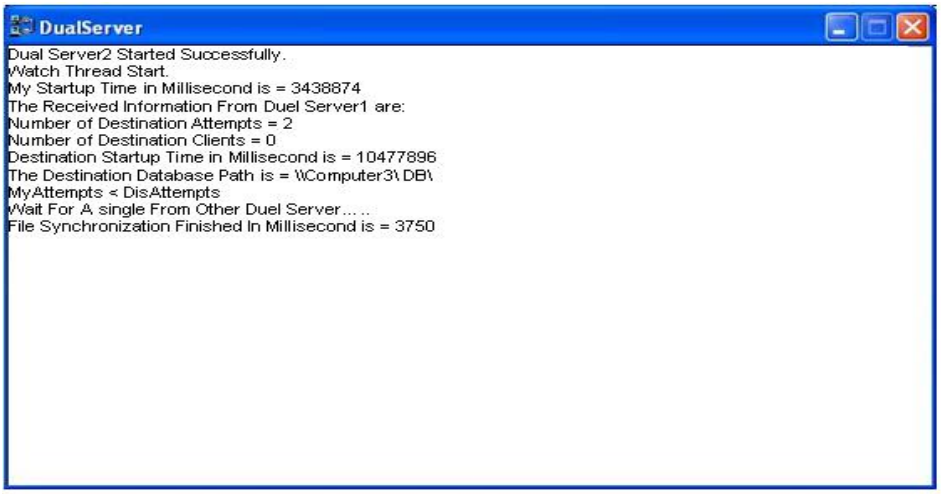

Figure (15) Dual_Server2

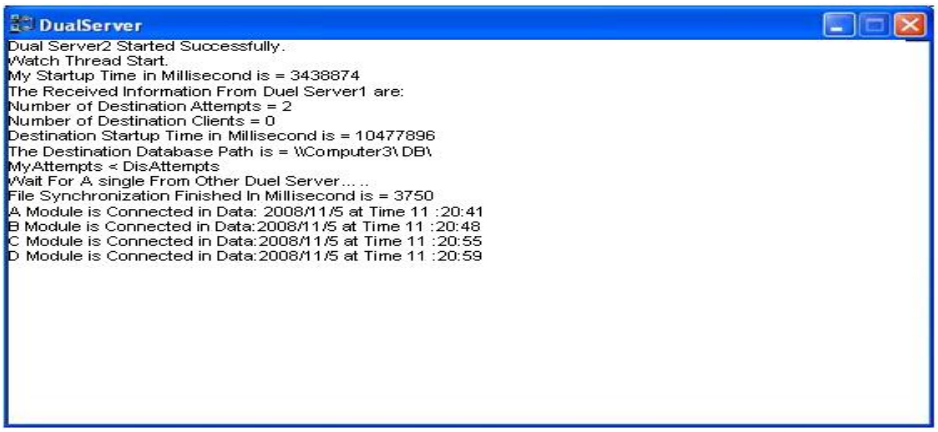

Figure (16) Clients Connect to Dual Server1

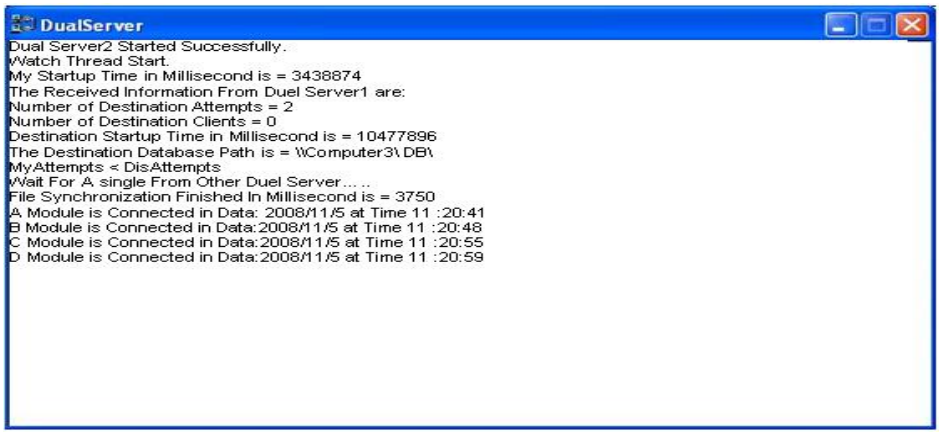

Figure (17) Clients Connect to Dual Server2

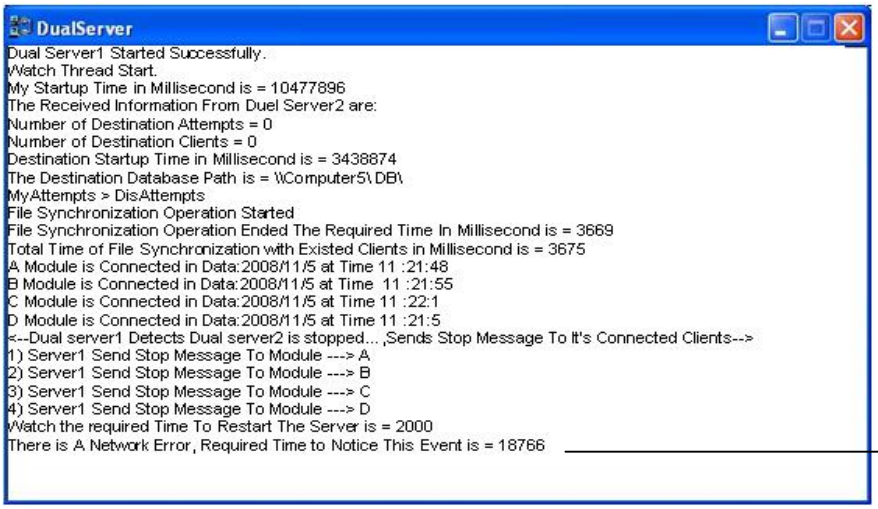

Figure (18) the Dual_Server1 Detect Dual_Server2 Fail

Note: The time is kept here to show the difference between central FTP and old Dual Server detection time see figure(10)

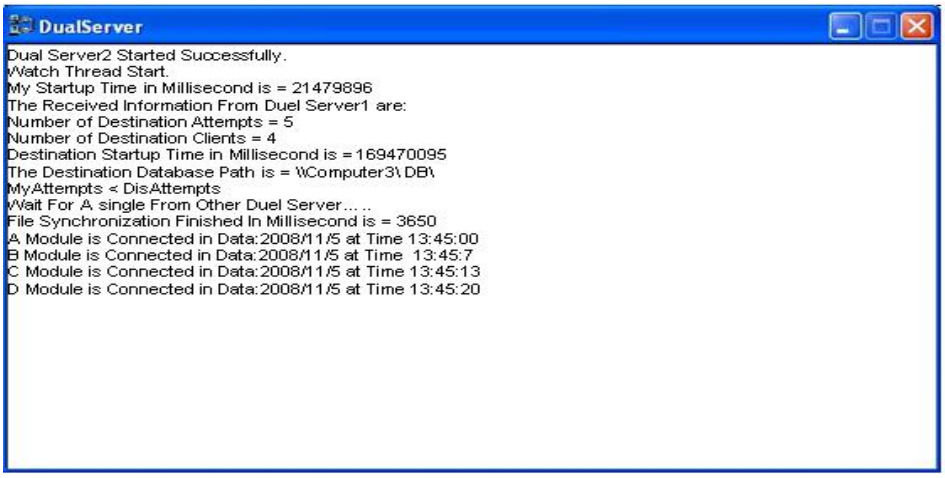

Figure (19) Dual_Server2 Reloaded by FTP on Its Backup Computer

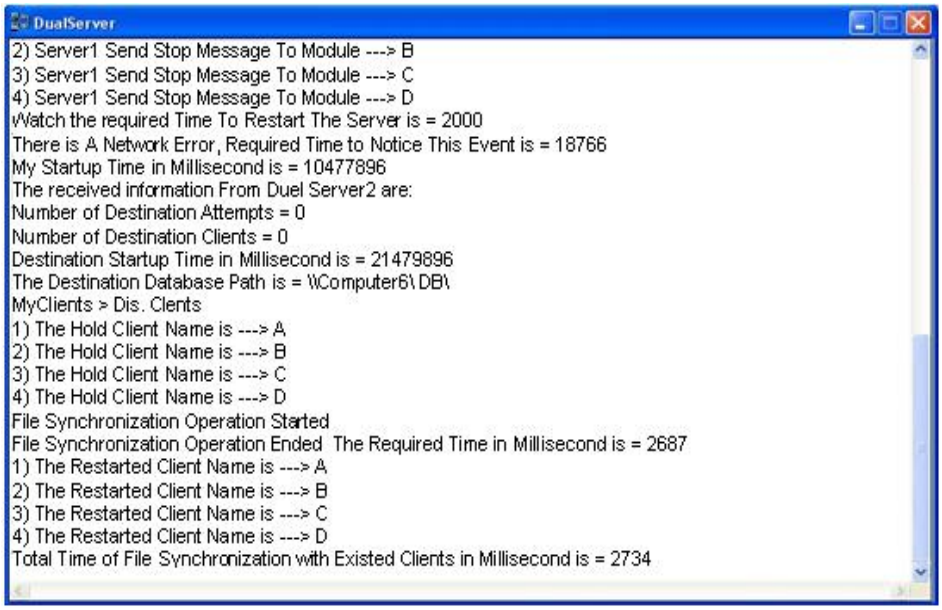

Figure (20) the Dual_Server1 Detect reload of Dual_Server2